SW-ELM: a summation wavelet extreme learning machine algorithm with *a priori* parameter initialization

Kamran Javed, Rafael Gouriveau[*], Noureddine Zerhouni

FEMTO-ST Institute, UMR CNRS 6174 - UFC / ENSMM / UTBM,
Automatic Control and Micro-Mechatronic Systems Department
24 rue Alain Savary, 25000 Besançon, France
firstname.lastname@femto-st.fr

Abstract

Combining neural networks and wavelet theory as an approximation or prediction models appears to be an effective solution in many applicative areas. However, when building such systems, one has to face parsimony problem, i.e., to look for a compromise between the complexity of the learning phase and accuracy performances. Following that, the aim of this paper is to propose a new structure of connectionist network, the Summation Wavelet Extreme Learning Machine (SW-ELM) that enables good accuracy and generalization performances, while limiting the learning time and reducing the impact of random initialization procedure. SW-ELM is based on Extreme Learning Machine (ELM) algorithm for fast batch learning, but with dual activation functions in the hidden layer nodes. This enhances dealing with non-linearity in an efficient manner. The initialization phase of wavelets (of hidden nodes) and neural network parameters (of input-hidden layer) is performed *a priori*, even before data are presented to the model. The whole proposition is illustrated and discussed by performing tests on three issues related to time-series application: an "input-output" approximation problem, a one-step ahead prediction problem, and a multi-steps ahead prediction problem. Performances of SW-ELM are benchmarked with ELM, Levenberg Marquardt algorithm for Single Layer Feed Forward Network (SLFN) and ELMAN network on six industrial data sets. Results show the significance of performances achieved by SW-ELM.

Keywords: Wavelet neural network, extreme learning machine, parameters initialization, activation functions, prediction accuracy

1. Introduction

Among several types of ANNs, feed forward neural networks (FFNN) have played an important role in research for different applications like pattern classification, complex non-linear mappings and other fields as well [1]. To instigate such structures and to achieve superior FFNN models for more complex applications, the combination of wavelet theory and learning ability, approximation properties of ANNs has resulted in the form of wavelet neural networks (WNNs) [2, 3]. In such an integration, the structure is based on a single layer feed forward network (a special type of FFNN) with wavelet activation functions in the hidden layer, which can also be called as wavenet [4]. According to literature, different models of WNNs have been proposed recently, Wang et al. [5] and Yamakawa et al. [6] used non-orthogonal wavelet function as a hidden node activation function in SLFN. Cao et al. [2] used wavelet with a bounded non-constant piecewise continuous function and proposed a new algorithm that had composite functions in hidden layer. Pourtaghi et al. [7] performed a thorough assessment on wavenet ability in comparison to SLFN and used different wavelets as activation functions to enhance the network performance. In order to tune such ANNs, various kinds of training schemes can be used, like support vector machines (SVM), evolutionary approaches, or simple back-propagation algorithm (BP) [8]. However, the main disadvantage of such learning methods is slow learning time. As for others issues, consider an example of the BP that suffers from network complexity, imprecise learning rate, over-fitting, presence of local minimum, etc. In order to overcome drawbacks of traditional learning methods, Huang et al. presented recently an efficient learning scheme for SLFN referred to as Extreme Learning Machine (ELM) [9]. Mainly, ELM algorithm for SLFN, randomly initializes parameters of hidden nodes and restricts the learning scheme to linear methods to analytically determine output weights for a suitable readout function. In brief, some prominent advantages of ELM are: (1) it is a one-pass algorithm, (2) it only requires tuning of one parameter i.e., hidden neurons, (3) it can work with wide range of activation functions including piecewise continuous functions. Among these advantages, learning speed is far superior to traditional methods like SVM and BP. However, random initialization of hidden node parameters

[*]Corresponding author, Tel.: +33 (0)3 81 40 27 96

may affect the performances of ELM [10], where, human choices, like the number of nodes or the type of activation functions, also have a great impact on the network's usefulness. Indeed, the last factor often plays a crucial role [1, 11]. Finally, due to such hurdles, performances of algorithms might not be as great as per expectations. Moreover, for practitioners, these problems may limit the application of an algorithm for real problems [12]. As an example, consider prediction of time series, that can be usually non-linear or chaotic by nature and requires to carefully choose an appropriate approach for prediction.
According to all this, the aim of this paper is to propose a new structure of connectionist network, the SW-ELM that enables good accuracy and generalization performances, while limiting the learning time and reducing the impact of random initialization procedure. The main contributions of this paper are as follows:

- *Fast learning of SW-ELM* - To achieve fast learning, ELM [9] is considered as a basis to train the SLFN. ELM is preferred mainly due to its features of increased applicability (to avoids drawbacks) as compared to conventional algorithms for ANNs.
- *Dual structure of SW-ELM* - To ensure good approximation capability while keeping a compact structure, a SLFN is integrated with wavelet such that each neuron in the hidden layer holds a dual activations (two different activation functions). By this configuration, the output from individual hidden node is averaged after transformations from dual activations [3].
- *Activation functions of SW-ELM* - To improve accuracy of the SW-ELM, a combination of a Morlet wavelet [2, 7] and an inverse hyperbolic sine (arcsinh) [13] is applied to each hidden node of the network. This enhances performances of hidden layers and enables dealing with non-linearity in an efficient manner.
- *Parameters initialization of SW-ELM* - To further optimize the network, the well known Nguyen Widrow procedure is applied to initialize hidden nodes weights and bias [14]. Also, parameters of wavelet activations functions are initialized thanks to an heuristic procedure based on the domain of each input [15].

The paper is organized as follows. Theoretical backgrounds of WNNs and ELM are given in section 2. This enables pointing out advantages and drawbacks from both models. On this basis, section 3 presents proposed SW-ELM with new structure and learning scheme. Performances of SWELM are benchmarked by performing tests on real datasets from industry (section 4). Three kinds of issues are considered: an "input-output" approximation problem, a one-step ahead prediction problem, and a multi-steps ahead prediction problem. Thorough comparisons with classical ELM and Levenberg Marquardt [16] for SLFN and ELMAN network are given. Finally, section 5 concludes this work and proposes some future aspects.

2. Backgrounds: WNN and ELM

2.1. The wavelet neural network WNN

2.1.1. Concept of wavelet

Wavelet theory is an outcome of multidisciplinary struggles, that brought together engineers, mathematicians and physicists [7]. The term wavelet means a "little wave". Mainly, a wavelet transform (WT) of continuous form behaves as a flexible time-frequency window, that shrinks when analyzing high frequency and spreads when low frequency behavior is observed [3]. The WT can be divided into two types, continuous wavelets transform (CWT) and discrete wavelets transform (DWT), formulated as follows:

$$CWT(a,b) = \frac{1}{\sqrt{a}} \int x\left(t\right) \Psi\left(\frac{t-b}{a}\right) \tag{1}$$

$$DWT(a,b) = \sum_i x\left(t\right) a_i^{-1/2} \Psi\left(\frac{t-b_i}{a_i}\right) \tag{2}$$

where Ψ is a wavelet function, and a and b are its corresponding scale (dilate) and translate (shift) factors.
It should be noted that CWT has the drawback of impracticality with digital computers, so, DWT is used in practice. Thus, the scale and translation factors from Eq. (2) are evaluated on a discrete grid of time scale to generate scaled and shifted daughter wavelets from a given mother wavelet Ψ. Additional details can be found in [3].

2.1.2. Structure of a WNN

An analogy can be found between the expression of the DWT and ANN output. Indeed, Eq. (2) can be seen as the output expression of a SLFN that would have an activation function Ψ for a hidden node, with a linear neuron in the output layer. Generally, such combination can be classified into two types. In the first case, the wavelet part is decoupled from network learning. In this manner, a signal is decomposed on some wavelets and its wavelet coefficients are furnished to a FFNN. In the second category, wavelet theory and FFNN are combined into a hybrid structure. The scope of this paper covers the latter category.

Let note n the number of inputs of a WNN, and $\tilde{N}$ the number of hidden nodes with wavelet functions (eg. morlet, see Fig. 1). According to this, the output of a WNN can be formulated as:

$$y = \sum_{k=1}^{\tilde{N}} v_k \Psi_k \left(\sum_{j=1}^{n} w_{kj} x_j \right) \tag{3}$$

$$\Psi_k\left(x\right) = \left|a_k\right|^{-1/2} \Psi\left(\frac{x-b_k}{a_k}\right) \tag{4}$$

where $x = x_1, x_2, ..., x_n$ depicts the input values, Ψ_k represents the family of daughter wavelets that are scaled and translated from a single mother wavelet function Ψ, a_k and b_k are the corresponding scale (dilate) and translation factors. Lastly, $w_k = [w_{k1}, w_{k2}, ..., w_{kn}]^T \in \Re^n$ is an input weight vector connecting the k^{th} hidden neuron to the input layer neurons, and v_k is the weight to connect the k^{th} neuron of hidden layer and the output.

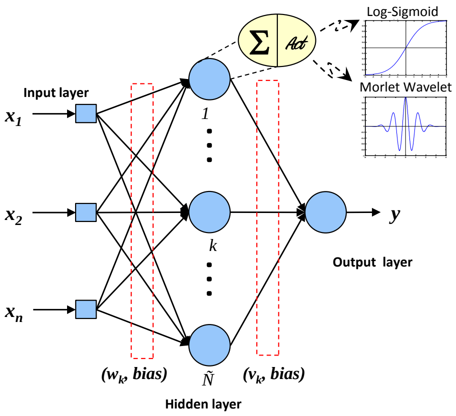

Figure 1: Single layer feed forward neural network

2.1.3. Issues and requirement

According to literature, the initialization of dilation and translation factors (a_k and b_k in Eq. (4)) is considered as a critical phase. Indeed, it is necessary to properly initialize these parameters for fast convergence of algorithm [2, 15]. As wavelets functions vanish rapidly, improper initialization may lead to the following issues:

- a wavelet can be too local for a very small dilation;
- improper translation may be out of interest domain.

Following that, random tuning of dilation and translation factors of wavelets is inadvisable, and parameters initialization should be based on the input domain. This aspect is taken into account in the proposed model (section 3).

2.2. Extreme Learning Machine for SLFN

2.2.1. Learning principles

ELM is a new learning scheme for SLFN that has been recently proposed by Huang et al. [9]. Almost all learning algorithms for SLFNs require adjustment of parameters that results dependence between different layers of parameters like, weights and biases. Therefore, many iterative tuning steps are required by traditional learning algorithms [17]. Where, the ELM algorithm avoids slow iterative learning procedure and only requires a one-pass operation to learn SLFN. This is mainly due to the fact that there is no bias for the output layer neuron (see Fig. 1). In brief, to initiate one-pass learning operation, the hidden node network parameters (weights and biases) are randomly generated without any prior knowledge or training procedure [2]. Consequently, the ELM turns into a system of linear equations and the unknown weights between the hidden and output layer nodes can be obtained analytically by only applying Moore-Penrose generalized inverse procedure [13, 18]. The learning scheme of ELM can be summarized in three steps (algorithm 1).

Algorithm 1 Learning scheme of an ELM

```
Assume
  - $n$ inputs, $m$ outputs, $\tilde{N}$ hidden nodes ($k = 1 \dots \tilde{N}$)
Require
  - $N$ learning data samples $(x_i, t_i)$ $(i = 1 \dots N)$
    $x_i = [x_{i1}, ..., x_{in}]^T \in \Re^n$, $t_i = [t_{i1}, ..., t_{im}]^T \in \Re^m$
  - An activation function $g(x)$
1: Randomly assign parameters of hidden nodes i.e., weights and bias $(w_k, bias_k)$.
2: Obtain the hidden layer output matrix $H$.
3: Find the output weight matrix $\beta$: $\beta = H^{\dagger}T$, where $H^{\dagger}$ represents the Moore-Penrose generalized inverse solution for the hidden layer output matrix $H$ [18].
```

2.2.2. Issues and requirement

As a synthesis, the ELM algorithm shows faster learning speed over traditional algorithms for SLFNs. For example, it learns faster than the Support Vector Machine by a factor of up to thousands. ELM does not suffer from problems like imprecise learning rate, presence of local minima etc. This underlines suitability of ELM for real complex applications that need fast prediction and response capabilities. In addition, the ELM algorithm requires less human involvement, because it does not have any control parameters to be manually tuned, except the number of hidden neurons, which shows its better applicability for real applications [10]. Finally, recent works confirm the advantages of ELM over earlier approaches for ANN [8, 17, 19, 20, 21]. Nevertheless, two key aspects should be pointed out.

- Random initialization of hidden node parameters may affect the performances of ELM [10], that also require high complexity of SLFN for improved performance. This may lead to ill-condition, which means that an ELM may not be robust enough to capture variations in data [22].
- Considering the network complexity issue, it is essential to carefully choose hidden neuron activation functions that show good ability to handle complexity, improve convergence of algorithm and result in a compact structure of network [13, 23, 24].

These aspects are taken into account in the proposed SW-ELM model (section 3).

3. Summation Wavelet-ELM model

3.1. Structure and mathematical perspective

As mentioned in the previous section, ELM is quiet efficient as compared to traditional methods to learn SLFNs. However, issues like parameters initialization, model complexity and choice of activation functions have to be carefully addressed for improved performance. Therefore, we propose the SW-ELM as shown in Fig. 2. The proposed

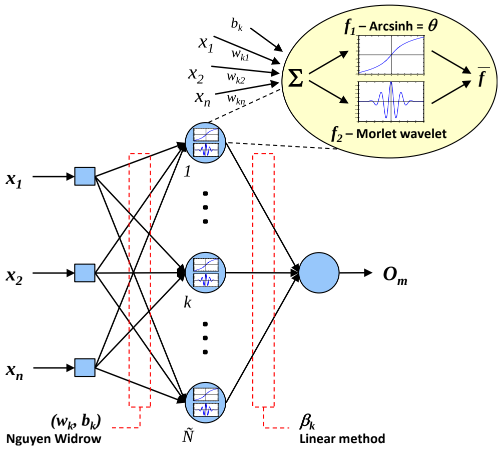

Figure 2: Structure of SW-ELM

structure takes advantages of WT and SLFN. Mainly, two significant modifications are performed in the hidden layer.

- Non-linear transformations are dealt in a better manner by using a conjunction of two distinct activation functions (f_1 and f_2) in each hidden node rather than a single activation function. The output from a hidden node is the average value after performing transformation from dual activations ($\bar{f} = (f_1 + f_2)/2$).
- To improve convergence of algorithm, an inverse hyperbolic sine ([13], Eq. (5)) and a Morlet wavelet ([2, 7], Eq. (6)) are used as dual activation functions.

$$f_1 = \theta(t) = arcsinh(t) = \int_0^t \frac{dx}{(1+x^2)^{1/2}} \tag{5}$$

$$f_2 = \psi(t) = cos(5t)\, e^{\left(-0.5t^2\right)} \tag{6}$$

Such a combination of activation functions makes the network more adequate to deal with dynamic systems. That is, the improved structure and proper choice of activation functions enhances the capability of the network to face low and high frequency signals simultaneously. As a consequence, the number of neurons required for hidden layer decreases and a compact structure is achieved [3, 13, 23]. According to modifications mentioned above, lets consider n and m the numbers of inputs and outputs, N the number of learning data samples (x_i, t_i), where $i \in [1 \dots N]$, $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in \Re^n$ and $t_i = [t_{i1}, t_{i2}, ..., t_{im}]^T \in \Re^m$, and $\tilde{N}$ the number of hidden nodes, each one with activation functions (f_1 and f_2). For each sample j, the output o_j is mathematically expressed as:

$$\sum_{k=1}^{\tilde{N}} \beta_k \bar{f}\left[(\theta, \psi)\left(w_k.x_j + b_k\right)\right] = o_j \ , j = 1, 2, ..., N \tag{7}$$

where $w_k = [w_{k1}, w_{k2}, ..., w_{kn}]^T \in \Re^n$, is an input weight vector connecting the k^{th} hidden neuron to the input layer neurons, $(w_k.x_j)$ is the inner product of weights and inputs, and $b_k \in \Re$ is the bias of k^{th} neuron of hidden layer. Also, $\beta_k = [\beta_{k1}, \beta_{k2}, ..., \beta_{km}]^T \in \Re^m$, is the weight vector to connect the k^{th} neuron of hidden layer and output neurons. Finally, $\bar{f}$ shows the average output from two different activation functions i.e., an inverse hyperbolic sine activation function θ and a Morlet wavelet activation function ψ. In order to minimize the difference between network output o_j and given target t_j, $\sum_{j=1}^{\tilde{N}} \|o_j - t_j\| = 0$, there exist β_k, w_k and b_k such that:

$$\sum_{k=1}^{\tilde{N}} \beta_k \bar{f}\left[(\theta, \psi)\left(w_k.x_j + b_k\right)\right] = t_j \ , j = 1, 2, ..., N \tag{8}$$

which can be expressed in matrix form as,

$$H_{avg}\beta = T \tag{9}$$

where H_{avg} is a $\left[N \times \tilde{N}\right]$ matrix expressed as,

$$H_{avg}\left(w_1, \dots, w_{\tilde{N}}, x_1, \dots, x_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}\right) =$$

$$\bar{f}(\theta, \psi) \begin{bmatrix} (w_1.x_1 + b_1) & \dots & (w_{\tilde{N}}.x_1 + b_{\tilde{N}}) \\ \vdots & \dots & \vdots \\ (w_1.x_N + b_1) & \dots & (w_{\tilde{N}}.x_N + b_{\tilde{N}}) \end{bmatrix} \tag{10}$$

and,

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_{\tilde{N}}^T \end{bmatrix}_{N \times m} \tag{11}$$

Finally, the least square solution of the linear system defined in Eq. (9), with minimum norm (magnitude) of output weights β is:

$$\hat{\beta} = H_{avg}^{\dagger} T = \left(H_{avg}^T H_{avg}\right)^{-1} H_{avg}^T T \tag{12}$$

where $H_{avg}^{\dagger}$ represents the Moore-Penrose generalized inverse solution for the hidden layer output matrix H_{avg} [18].

3.2. Learning scheme of SW-ELM

Main learning phase derives from Eq. 10 and 12. However, according to issues and requirements related to WNN and ELM presented in sections 2.1.3 and 2.2.2, it is desired to take care of parameters initialization task and to provide a better starting point to algorithm. Two types of parameters have to be considered: the ones from the wavelets (dilation and translation factors) and the ones from the SLFN (weights and bias for input to hidden layer nodes). Details of the learning scheme are given in algorithm 2, and an outline of the main step is synthesized herefater.

Initializing wavelet parameters. To initialize wavelet dilation and translation parameters (a_k and b_k in Eq. (4)) before the learning phase, a heuristic approach is applied to generate daughter wavelets from a mother wavelet function (in our case, a Morlet wavelet). Dilation and translation values are adapted by considering the domain of

Algorithm 2 Learning scheme of the SW-ELM

```
Require  - $N$ learning data samples $(x_i, t_i)$, $n$ inputs $(j = 1 \dots n)$, $\tilde{N}$ hidden nodes $(k = 1 \dots \tilde{N})$
         - An inverse hyperbolic sine and a Morlet wavelet activation functions ($\theta$ and $\psi$)
Ensure   - Initialize weights and bias from SLFN, initialize Morlet parameters
         - Find output weights matrix $\beta$ to minimize the difference between the network' outputs and the targets
SW-ELM learning procedure
Initialization of wavelet parameters
    - Define the input space domain intervals
        - Compute $[x_{jmin} \ ; \ x_{jmax}]$: {domain containing the input item $x_j$ in all observed samples}
    - Define dilation and translation parameters per domain
        - Compute $d_{kj} = 0,2 \times [x_{jmax} - x_{jmin}]$: {temporal dilatation parameter for input item $x_j$}
        - Compute $m_{kj} = [x_{jmin} + x_{jmax}]/2$: {temporal translation parameter for input item $x_j$}
    - Initialize Morlet parameters ($a_k$ and $b_k$)
        - Compute $a_k = mean(d_{kj})_{j=1 \dots n}$: {dilatation factor}
        - Compute $b_k = mean(m_{kj})_{j=1 \dots n}$: {translation factor}
Initialization of weights and bias parameters by Nguyen Widrow (NW) approach
    - Initialize small (random) input weights $w_{k(old)}$ in $[-0.5 \ ; \ +0.5]$: {weights from input nodes to hidden nodes}
    - Adjust weights parameters by applying NW approach
        - Compute $\beta_{factor} = C \times \tilde{N}^{\frac{1}{n}}$: {$C$ is a constant $\leq 0.7$}
        - Compute $w_{k(new)} = \beta_{factor} \times \frac{w_{k(old)}}{\|w_{k(old)}\|}$: {normalized weights}
    - Initialize bias values $b_k$
        - $b_k$ = random number between $-\beta_{factor}$ and $+\beta_{factor}$
Adjust linear parameters: the ones from the hidden to the output layers
    - Obtain hidden layer output matrix $H_{avg}$ using Eq. 10
    - Find the output weight matrix $\hat{\beta}$ in Eq. 12 by applying Moore-Penrose generalized inverse procedure
```

the input space where wavelet functions are not equal to zero [15]. Besides that, wavelet function with a small dilation value are low frequency filters, whereas increasing dilation factors the wavelet behave as high frequency filter [3]. Finally, the effects of random initialization of wavelet parameters are avoided, and the initialization guarantees that wavelet function stretches over the whole input domain [15].

Initializing weights and bias. The hard random parameters initialization step is substituted by well known Nguyen Widrow (NW) procedure to initialize weights and bias [14]. The intent is to provide a better starting point for learning. NW method is a simple alteration of hard random initialization that aims at adjusting input weights intervals and hidden bias according to the input-hidden layer topology. It has been argued that NW method has shown improved performances over others methods for random parameters initialization of ANNs [25].

4. Experiments and discussion

4.1. Outline: aim of tests and performance evaluation

The aim of this part is to demonstrate enhanced performances of the proposed SW-ELM, that is benchmarked with the ELM, Levenberg-Marquardt (LM) algorithm for SLFN and ELMAN network. For simulation purpose, a sigmoid function is used as hidden node activation function for ELM, LM-SLFN and ELMAN network, whereas for SW-ELM, dual activations are used: an inverse hyperbolic sine function and a Morlet wavelet (Eq. (5) and (6)). Note that, number of hidden neurons for each model are assigned using trial and error approach, which obviously could not guarantee optimal structure.

Simulations are carried on six real datasets from industry as shown in Table 1, where information about model inputs, outputs and training, testing samples are also mentioned (see references for further details on the datasets). Three kind of issues related to time-series application are considered: an "input-output" approximation problem, a one-step ahead prediction problem, and a multi-steps ahead prediction problem.

To compare performances, three criteria are considered.

1. Computation time to learn the dataset for a single trial ($Time$).
2. Model fitting accuracy is judged by the coefficient of determination ($R2$) that should be close to 1 and coefficient of variation of the Root Mean Squared Error ($CVRMSE$) that should be as low as possible (close to 0 - it is often expressed in percentage). Both measures are unitless and are indicative of model fit, but each define model fit in two different ways: $R2$ evaluates the variability in the actual values explained by the model, where $CVRMSE$ evaluates the relative closeness of the predictions to the actual values.
3. Network complexity is reflected by the number of hidden neurons ($Hid - nodes$).

Table 1: Specification of datasets to benchmark performances for time series application

Data	Short description	Input	Output	Train	Test
Pump [26]	Condition monitoring	Root mean square Variance	Fault code	73 (samples)	19 (samples)
CNC [27]	Condition monitoring	Max force, Cutting amp. Amp. Ratio, Avg. force	Tool wear	C33, C09, C18 (450 samples)	C18 (165 samples)
Ind. Dryer [28]	Predict temperature	Fuel flow, Fan speed Flow raw material Bulb temp. y_t	Bulb temp. y_{t+1}	500 (samples)	367 (samples)
Hair dryer [29]	Predict temperature	Voltage of device x_t Air temp. y_t	Air temp. y_{t+1}	500 (samples)	500 (samples)
NN3 [30]	Time series forecast	Time series (4 reg.) $(x_t, x_{t-1}, x_{t-2}, x_{t-3})$	Same time series $x_{t+1 \to t+18}$	51, 54, 56, 58, 60, 61, 92, 106	All series (18 samples)
Turbofan [31]	Predict degradation	Degradation series 3 reg. (x_t, x_{t-1}, x_{t-2})	Same series $x_{t+1 \to t+H}$	90 engines	5 engines $H \in [103, 283]$

With each approach (i.e., ELM, SWELM, LM-SLFN, ELMAN), *50* simulations were performed on a particular dataset and the best results are summarized in Table 2.

4.2. First issue: an approximation problem

In case of approximation or estimation problem, real datasets from two different degrading machineries were used for condition monitoring task. The first dataset was from a *Carnallite surge tank pump*, which was used to approximate fault codes, where the second dataset was from a *Computer Numerical Control milling machine* to estimate wear of degrading cutting tools (Table 1). Results in Table 2 show that SW-ELM for both test datasets performs best approximations, with a compact structure, and it requires less learning time.

1. ELM has clear superiority of fast learning times ($5.8e^{-004}$ and $5.0e^{-004}$ sec) for both datasets. In addition, SW-ELM and ELM can learn much rapidly as compared to LM-SLFN or ELMAN network. This is mainly due to the advantages of single-step learning phase of ELM based models, whereas LM-SLFN and ELMAN network required additional 50 epochs for the *Pump* dataset and 10 epochs for *CNC* dataset to achieve better accuracy performances.
2. The accuracy of SW-ELM show a best fitting among all methods for both datasets as measured by $R2$ i.e., ($R2$=0.96 and $R2$=0.92). Comparative plots representing approximation of fault codes and tool wear estimation are shown in Fig. 3 and Fig. 4, where model fitting errors are also depicted for a better understanding of results. Note that, for more clarity, plots of only two methods with higher accuracies are compared.
3. Lastly, when considering model complexity factor, the number of hidden neurons required by LM-SLFN and ELMAN network appears to be twice (i.e. 30 hidden nodes) as compared to ELM based models when applied to Pump dataset. However, same network complexity of 4 hidden nodes is sufficient for all methods with CNC dataset.

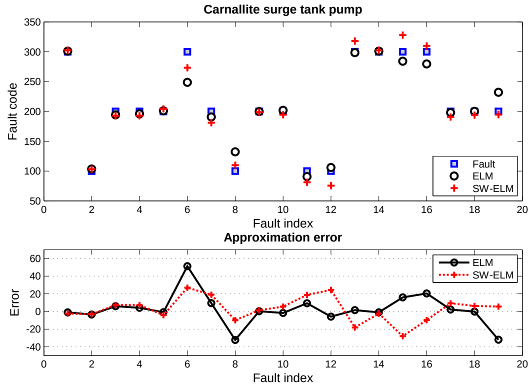

Figure 3: Fault code approximations and corresponding errors

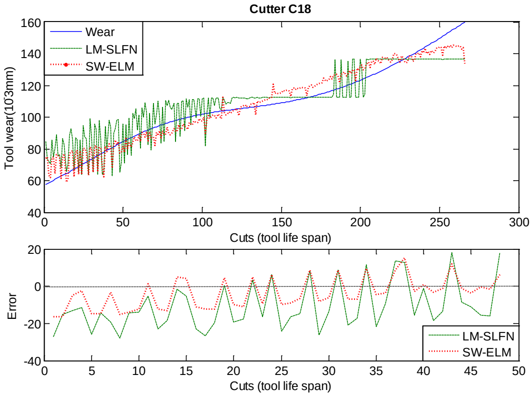

Figure 4: Tool wear estimation and corresponding errors

4.3. Second issue: one-step ahead prediction problem

In case of one-step ahead prediction issue the datasets used, were from an *industrial dryer* and a *mechanical hair*

Table 2: Comparison of model performances

	Approximation: *Pump*				**Approximation: *CNC***			
Method	Hid-node	Train time (sec)	Epoch	R2	Hid-node	Train time (sec)	Epoch	R2
SW-ELM	**15**	6.5e-004	–	**0.96**	**4**	7.7e-004	–	**0.92**
ELM	15	**5.8e-004**	–	0.94	4	**5.0e-004**	–	0.77
LM-SLFN	30	1.02	50	0.79	4	0.22	10	0.80
ELMAN	30	8.88	50	0.81	4	0.21	10	0.77
	1-step prediction: *Ind. Dryer*				**1-step prediction: *Hair dryer***			
Method	Hid-node	Train time (sec)	Epoch	R2	Hid-node	Train time (sec)	Epoch	R2
SW-ELM	**20**	0.0024	–	**0.85**	**4**	6.1e-004	–	**0.944**
ELM	20	**0.0012**	–	0.66	4	**3.4e-004**	–	**0.944**
LM-SLFN	30	1.03	50	0.81	4	0.21	10	0.9434
ELMAN	30	8.9	50	0.80	4	0.20	10	0.9434
	Multi-step prediction: *NN3* (avg.)				**Multi-step prediction: *Turbofan*** (avg.)			
Method	Hid-node	Train time (sec)	Epoch	CVRMSE	Hid-node	Train time (sec)	Epoch	CVRMSE
SW-ELM	**30**	0,0014	–	**10.83%**	**3**	0.006	–	**0.042%**
ELM	30	**5.5e-004**	–	11.06%	3	**0.004**	–	0.0578%
LM-SLFN	30	0.20	10	11.51%	3	0.72	10	0.0570%
ELMAN	30	0.45	10	**10.83%**	3	0.75	10	0.0570%

dryer, that were applied to predict temperature variations (Table 1). In order, to judge model performances, same criteria are used. Comparative performances from all models are summarized in Table 2. Again, results clearly indicate that SW-ELM shows improved performances over ELM, LM-SLFN and ELMAN network.

1. ELM can still perform faster learning (0.0012 and $3.4e^{-004}$ sec) with both datasets, even if the learning data size is increased (500 samples). Like in previous issue, the learning times of SW-ELM are still close to the ones of ELM.
2. The accuracy indicator of SW-ELM shows a higher prediction performance with an accuracy $R2 = 0.85$ with dataset of an *industrial dryer*. However the accuracy of SW-ELM and ELM are the same ($R2 = 0.944$) in the case of second data of a *mechanical hair dryer*. It should be noted that, in the second test all methods showed good prediction performances. As for illustration, comparative plots of predictions with only two methods are shown in Fig. 5 and Fig. 6 by considering model accuracy, where predictions errors are also depicted. In Fig. 5, one can note that SW-ELM catches better non-linearity from the signal since the error is quite constant among all predictions.
3. ELM based models show a more compact structure (20 hidden nodes) as compared to LM-SLFN and ELMAN network (30 hidden nodes) when applied to *industrial dryer* data. However, again small network complexity of 4 hidden nodes is sufficient for all methods with the second data of *mechanical hair dryer*.

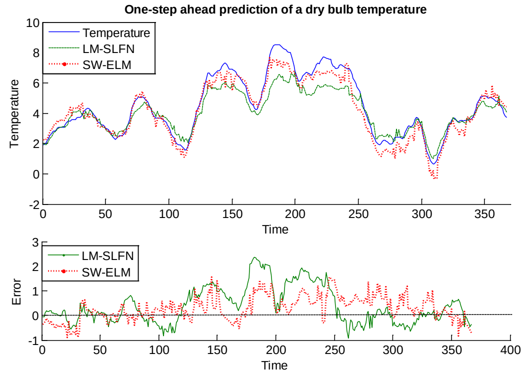

Figure 5: 1-step ahead pred. of bulb temp. and corresponding errors

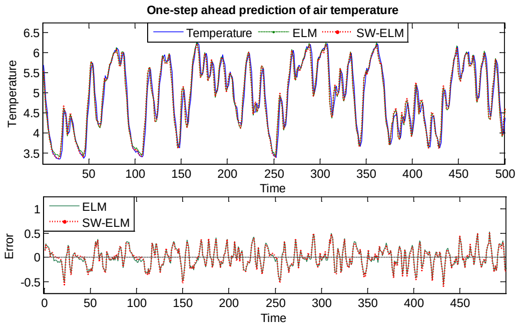

Figure 6: 1-step ahead pred. of Air temp. and corresponding errors

4.4. Third issue: multi-steps ahead prediction problem

4.4.1. Multi-steps ahead prediction models

Multi-steps ahead prediction (MSP) modeling by using neural networks can be achieved in different manners. This can not be fully addressed in this paper, but one can refer to [32] for more details. Here, MSP are met by using the "Iterative" approach that is the most common and the simplest to implement.

Consider a univariate time series $\mathbf{X}_t$, i.e. a sequence of values describing an observation made at equidistant intervals $\mathbf{X_t} = \{x_1, x_2, \ldots, x_t\}$. The MSP problem consists of estimating a set of future values of the time series $\hat{\mathbf{X}}_{t+1\rightarrow t+H}$, where H states for the final horizon of prediction. For that purpose, the Iterative approach utilizes a single neural network that is tuned to perform a one-step ahead prediction $\hat{x}_{t+1}$. This estimated value is used as one of the regressors of the model to estimate another, and the operation is repeated until the estimation of $\hat{x}_{t+H}$ (see Fig. 7).

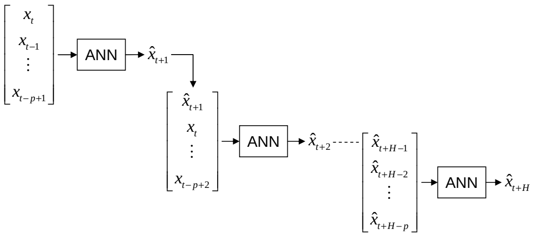

Figure 7: Iterative approach for multi-steps ahead predictions [32]. Parameter p states for the number of regressors

4.4.2. Results and discussion

In case of MSP problem, again two data sets were applied to test model performances. The first test data, were from *NN3 competition*, where horizon of prediction was last 18 values of each time series for test. The second dataset were from NASA data repository that were composed of run to failure sensor measurements from degrading *Turbofan engines*, where the horizon of prediction was from a critical time **tc** upto end of degradation (see Table 1, we used data record $train - FD001.txt$ to train and test the models). Similarly, like in previous issues, 50 trials are performed and best results are presented.

The MSP accuracy of each model is assessed by computing Coefficient of Variation of the Root Mean Squared Error ($CVRMSE$) that enables evaluating relative closeness of predictions, which should be as low as possible. Whereas, computational time (for a single trial) and network complexity are evaluated similarly like previous cases.

As for illustration, a comparative plot with two models showing higher accuracies of MSP with *NN3 competition* data for time series 61 is shown in Fig. 8. Averaged performances of prediction models for all randomly selected time series (51, 54, 56, 58, 60, 61, 92, 106) are summarized in Table 2. In case of *Turbofan engines* dataset, the comparative plot of two models on test-engine 5 is shown in Fig. 9, where SW-ELM explicitly shows good prediction capability over long horizon. Averaged performances of prediction models for all tests are summarized in Table 2.

1. Among all methods, ELM requires less computation time for learning with both datasets (i.e., $5.5e^{-004}$ and 0.004 sec). As explained before, this is mainly due to the one-pass learning phase, where SW-ELM also shows rapid learning behavior closer to ELM.
2. Most importantly, accuracy indicator ($CVRMSE$) of SW-ELM is highest in comparison to others models for both datasets (i.e., *NN3 competition* and *Turbofan engines*). A synthetic but explicit view of that is given in Fig. 10 and Fig. 11. Note that such performances of SW-ELM are obtained with the same structure like other methods (see point 3).
3. The network complexity of all models was the same with both datasets (i.e., 30 hidden nodes and 3 hidden nodes). Note that for all test on 5 *Turbofan engines*) even with more complex models of ELM, LM-SLFN and ELMAN network poor prediction performances were obtained, therefore the only complexity of 3 hidden nodes was compatible with the data.

Finally, again SW-ELM enables a good compromise among model accuracy, learning time and network complexity.

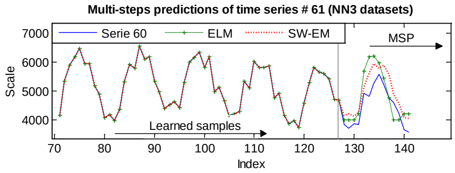

Figure 8: Multi-steps ahead predictions of time series 61 (NN3)

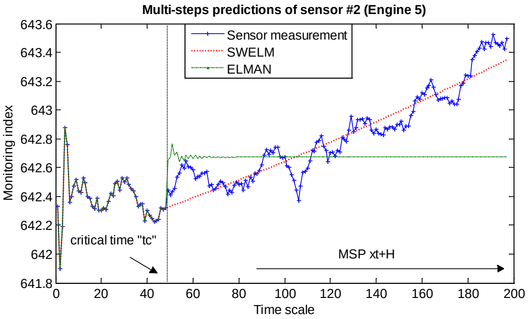

Figure 9: Multi-steps ahead predictions of sensor 2 (Engine 5)

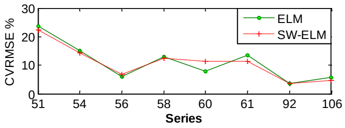

Figure 10: Accuracy of MSP for 8 NN3 series

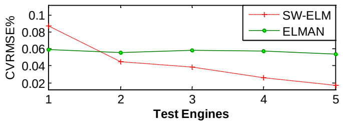

Figure 11: Accuracy of MSP for 5 Turbofan engines

5. Conclusion

In this paper, a new structure of neural network is proposed, the SW-ELM, which is based on ELM algorithm for fast learning, but with dual activation functions in the hidden layer. The output from each hidden node is the average value after transformations from an inverse hyperbolic sine function and a Morlet wavelet function. This enhances dealing with non-linearity in an efficient manner. Also, the learning scheme of SW-ELM is improved (w.r.t classical ELM), where wavelets and other parameters of hidden nodes are adjusted *a priori* to learning. The whole enables good accuracy and generalization performances, while limiting the learning time and reducing the impact of random initialization procedure. For practical problems, SW-ELM avoids long "test and error" procedure and is therefore suitable for real cases where few prior knowledge is available or huge datasets have to be processed. The performances of proposed SW-ELM are benchmarked with ELM, Levenberg Marquardt for SLFN, and ELMAN network for three types of prediction problems: input-output approximation, one-step ahead prediction, and multi-steps ahead predictions. In all cases, SW-ELM shows enhanced performance to face parsimony problem.

References

[1] G. Daqi, Y. Genxing, Influences of variable scales and activation functions on the performances of multilayer feedforward neural networks, Pattern Recognition 36 (4) (2003) 869 – 878.

[2] J. Cao, Z. Lin, G.-B. Huang, Composite function wavelet neural networks with extreme learning machine, Neurocomputing 73 (7-9) (2010) 1405 – 1416.

[3] A. Banakar, M. F. Azeem, Artificial wavelet neural network and its application in neuro-fuzzy models, Applied Soft Computing 8 (4) (2008) 1463 – 1485.

[4] X. Li, H. Zeng, J. H Zhou, S. Huang, T. B Thoe, K. C Shaw, B. S Lim, Multi-modal sensing and correlation modelling for condition-based monitoring in milling machine., SIMTech technical reports 8 (1) (2007) 50 – 56.

[5] W. Ting, Y. Sugai, A wavelet neural network for the approximation of nonlinear multivariable function, in: Proceedings of 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics, IEEE SMC'99, Vol. 3, 1999, pp. 378 –383.

[6] T. Yamakawa, E. Uchino, T. Samatsu, Wavelet neural networks employing over-complete number of compactly supported non-orthogonal wavelets and their applications, in: Proceedings of 1994 IEEE World Congress on Computational Intelligence, Vol. 3, 1994, pp. 1391 – 1396.

[7] A. Pourtaghi, M. Lotfollahi-Yaghin, Wavenet ability assessment in comparison to ann for predicting the maximum surface settlement caused by tunneling, Tunnelling and Underground Space Technology 28 (2012) 257 – 271.

[8] R. Rajesh, J. S. Prakash, Extreme learning machines - a review and state-of-the-art, International journal of wisdom based computing 1 (1) (2011) 35 – 49.

[9] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: A new learning scheme of feedforward neural networks, in: Proceedings of the International Joint Conference on Neural Networks IJCNN, Budapest, Hungary, 2004.

[10] A. U. Bhat, S. S. Merchant, S. S. Bhagwat, Prediction of melting points of organic compounds using extreme learning machines, Industrial & Engineering Chemistry Research 47 (3) (2008) 920 – 925.

[11] G. da S. Gomes, T. Ludermir, L. Lima, Comparison of new activation functions in neural network for forecasting financial time series, Neural Computing and Applications 20 (3) (2011) 417 – 439.

[12] R. Singh, S. Balasundaram, Application of extreme learning machine method for time series analysis, International Journal of Computer Systems Science and Engineering 2 (4) (2007) 256 – 262.

[13] M.-B. Li, G.-B. Huang, P. Saratchandran, N. Sundararajan, Fully complex extreme learning machine, Neurocomputing 68 (2005) 306 – 314.

[14] D. Nguyen, B. Widrow, Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights, in: Proceedings of the International Joint Conference on Neural Networks IJCNN, San Diego, CA, USA, 1990.

[15] Y. Oussar, G. Dreyfus, Initialization by selection for wavelet network training, Neurocomputing 34 (1-4) (2000) 131 – 143.

[16] M. T. Hagan, M. B. Menhaj, Training feedforward networks with the marquardt algorithm, Neural Networks, IEEE Transactions on 5 (6) (1994) 989–993.

[17] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (2006) 489 – 501.

[18] C. R. Rao, S. K. Mitra, Generalized Inverse of Matrices and its Applications, John Wiley and Sons, New York, 1972.

[19] G.-B. Huang, D. H. Wang, Y. Lan, Extreme learning machines: a survey, International Journal of Machine Learning and Cybernetics 2 (2) (2011) 107 – 122.

[20] V. Malathi, N. Marimuthu, S. Baskar, K. Ramar, Application of extreme learning machine for series compensated transmission line protection, Engineering Applications of Artificial Intelligence 24 (5) (2011) 880 – 887.

[21] Y. Wang, F. Cao, Y. Yuan, A study on effectiveness of extreme learning machine, Neurocomputing 74 (16) (2011) 2483 – 2490.

[22] G. Zhao, Z. Shen, C. Miao, Z. Man, On improving the conditioning of extreme learning machine: a linear case, in: Proceedings of 7th International Conference on Information, Communications and Signal processing, ICICS'09, Piscataway, NJ, USA, 2009.

[23] H. A. Jalab, R. W. Ibrahim, New activation functions for complex-valued neural network, International Journal of the Physical Sciences 6 (7) (2011) 1766 – 1772.

[24] G.-B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (16-18) (2008) 3460 – 3468.

[25] A. Pavelka, A. Prochazka, Algorithms for initialization of neural network weights, 2004, http://dsp.vscht.cz/konference_matlab/matlab04/pavelka.pdf.

[26] M. Samhouri, A. Al-Ghandoor, S. A. Ali, I. Hinti, W. Massad, An intelligent machine condition monitoring system using time-based analysis: Neurofuzzy versus neural network, Jordan Journal of Mechanical and Industrial Engineering 3 (4) (2009) 294 – 305.

[27] J. Zhou, X. Li, O.P. Gan, S. Han, W.K. Ng, Genetic algorithms for feature subset selection in equipment fault diagnostics, Engineering Asset Management 10 (2006) 1104 – 1113.

[28] Industrial dryer
ftp://ftp.esat.kuleuven.ac.be/sista/data/process_industry.

[29] Hair dryer
ftp://ftp.esat.kuleuven.ac.be/sista/data/mechanical.

[30] NN3 forecasting competition

http://www.neural-forecasting-competition.com/nn3/.

[31] A. Saxena, K. Goebel, D. Simon, N. Eklund, Damage propagation modeling for aircraft engine run-to-failure simulation, in: Prognostics and Health Management, 2008. International Conference on, IEEE, 2008, pp. 1–9.

[32] R. Gouriveau, N. Zerhouni, Connexionist-systems-based long term prediction approaches for prognostics, IEEE Transactions on Reliability 61 (4) (2012) 909 – 920.

Kamran Javed received his BS. computer engineering degree from COMSATS University Abbottabad (Pakistan) in 2006. He was honored by a scholarship to complete his Masters in computer engineering from COMSATS, in 2009. During his MS. he also worked as a research associate to develop Power system Planning Suit for WAPDA organization Pakistan, which mainly works for Hydel Power and Water Sector projects. In 2009 he joined COMSATS for a lecturer position in electrical engineering department. Currently he performs his doctoral studies at FEMTO-ST institute, in Besançon (France) that is funded by Ministry of France since 2011. His research activities are concerned with developing prognostics systems using methods based on artificial intelligence, in particular connectionist methods like extreme learning machine networks, and advanced neuro-fuzzy methods.

Rafael Gouriveau received his engineering degree from National Engineering School of Tarbes (ENIT) in 1999. He then got his MS (2000), and his Ph.D. in Industrial Systems in 2003, both from the Toulouse National Polytechnic Institute (INPT). During his PhD, he worked in the field of risk management and dependability analysis. In September 2005, he joined the National Engineering Institute in Mechanics and Microtechnologies of Besançon (ENSMM) as Associate Professor. His main teaching activities are concerned with production, maintenance, manufacturing, and informatics domains. As for investigation, Rafael Gouriveau is with the AS2M (Automatic Control and Micro-Mechatronic Systems) department of FEMTO-ST Institute. He is currently at the head of the PHM team in this department. His research interests are in the development of industrial prognostics systems using connexionist approaches like neuro-fuzzy methods, and the investigation of reliability modeling using possibility theory. He is also the scientific coordinator of PHM research axes from the FCLAB (Fuel Cell Lab) Research Federation (CNRS), devoted to Fuel Cell Systems.

Noureddine Zerhouni received his engineering degree from National Engineers and Technicians School of Algiers (ENITA) in 1985. After a short period in industry as an engineer, he received his Ph.D. Degree in Automatic Control from the Grenoble National Polytechnic Institute in 1991. In September 1991, he joined the National Engineering School of Belfort (ENIB) as Associate Professor. At this time, his main research activity is concerned with modeling, analysis, and control of manufacturing systems. Since September 1999, Noureddine Zerhouni has been a Professor at the national high school of mechanics and microtechniques of Besançon. As for investigation, he is with AS2M (Automatic Control and Micro-Mechatronic Systems) department of Femto-st Institute. His main research activities are concerned with intelligent maintenance systems, and e-maintenance. Professor Noureddine Zerhouni has been and is involved in various European and National projects on intelligent maintenance systems like the FP5 European Integrated Project of the ITEA program (Information Technology for European Advancement) PROTEUS, NEMOSYS (Naval E-Maintenance Oriented SYStem) with DCNS, and AMIMAC-FAME (Reliability Improvement of Embedded Machines) with ALSTOM.